Srinivasan Kamalanathan.

Employee Management Project developed using RESTful service using Java, Spring Boot, Maven, JPA, and RDBMS (MySQL) using all appropriate exception handlins

Requested Operations:

* Retrieve all Employees - (“/employees”)
* Get details of specific Employees -(“/employees/{id}”)
* Delete an Employee - (“/employees/{id}”)
* Create a new Employee - (“/employees”)
* Update Employee details - (“/employees/{id}”)

Employee Entity:

* FIRSTNAME
* LASTNAME
* EMAILADDRESS
* PHONE
* BIRTHDATE
* JOB-TITLE
* DEPARTMENT
* LOCATION
* STARTDATE
* EMPLOYEEID
* MANAGER-REPORTING

1.Contract (Swagger specification) first to generate the code

Contract (Swagger specification) first to generate the code approach, such as Swagger (Open API), allows you to define the API specifications before generating any code. The process includes writing the swagger specification, generate code, implementation, testing and documentation.

2 Code first and then generates the contract (Swagger specification)

Project development start with API using a Code-First approach involves writing the API code first and then generating the API contract (Swagger specification) based on the implemented code. This approach is sometimes preferred when developers want more flexibility in the implementation and are comfortable defining the API interface directly in code. The process includes writing the Api code, annotate with metadata, swagger specification, refine and review, and implementation, testing and documentation

1. writing the code, we can use tools like swagger2 to generate the Swagger specification. These tools scan the codebase, extract information from annotations like @GetMapping, @PostMapping, etc., and generate the Swagger document accordingly.
2. Once the Swagger specification is generated, it can be used for documentation, testing, and client code generation, providing a clear and consistent contract for the API.

3.jUnit/Postman Test Collection

Create both jUnit tests and a Postman test collection to verify that the API endpoint returns the expected response. In this scenario, jUnit is utilized to verify methods within the code, while Postman used to validate the results using URLs and various HTTP methods such as GET, POST, PUT, and DELETE operations